How can I calculate the cosine similarity in Python?

Authored by

stats writer

April 21, 2024

Cosine similarity is a mathematical measure used to determine the similarity between two vectors in a multi-dimensional space. In the context of Python, it is a popular technique for comparing and analyzing data sets. To calculate the cosine similarity in Python, one can use the built-in functions and libraries such as NumPy and SciPy. These libraries provide efficient implementations of the cosine similarity formula, allowing users to easily compute the similarity between two vectors. By using these tools, one can accurately determine the degree of resemblance between two data sets, making it a valuable tool for various data analysis tasks.

Calculate Cosine Similarity in Python

Cosine Similarity is a measure of the similarity between two vectors of an inner product space.

For two vectors, A and B, the Cosine Similarity is calculated as:

Cosine Similarity = ΣAiBi / (√ΣAi2√ΣBi2)

This tutorial explains how to calculate the Cosine Similarity between vectors in Python using functions from the NumPy library.

Cosine Similarity Between Two Vectors in Python

The following code shows how to calculate the Cosine Similarity between two arrays in Python:

from numpy import dot
from numpy.linalgimport norm

```
#define arrays
a =
b =

#calculate Cosine Similarity
cos_sim = dot(a, b)/(norm(a)*norm(b))

cos_sim

0.965195008357566
```

The Cosine Similarity between the two arrays turns out to be 0.965195.

Note that this method will work on two arrays of any length:

```
import numpy as np
from numpy import dot
from numpy.linalgimport norm

#define arrays
a = np.random.randint(10, size=100)
b = np.random.randint(10, size=100)

#calculate Cosine Similarity
```

cos_sim = dot(a, b)/(norm(a)*norm(b))

cos_sim

0.7340201613960431

However, it only works if the two arrays are of equal length:

```
import numpy as np
from numpy import dot
from numpy.linalgimport norm

#define arrays
a = np.random.randint(10, size=90) #length=90
b = np.random.randint(10, size=100) #length=100#calculate Cosine Similarity
cos_sim = dot(a, b)/(norm(a)*norm(b))
```

cos_sim

ValueError: shapes (90,) and (100,) not aligned: 90 (dim 0) != 100 (dim 0)

Notes

1. There are multiple ways to calculate the Cosine Similarity using Python, but as this Stack Overflow thread explains, the method explained in this post turns out to be the fastest.

2. Refer to this Wikipedia page to learn more details about Cosine Similarity.